

25th International Meshing Roundtable (IMR25)

# Feature-Aligned Surface Parameterization Using Secondary Laplace Operator and Loop Subdivision

Kangkang Hu<sup>a</sup>, Yongjie Jessica Zhang<sup>a,\*</sup>, Xinge Li<sup>b</sup>, Guoliang Xu<sup>b</sup><sup>a</sup>*Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, USA*<sup>b</sup>*LSEC, Institute of Computational Mathematics, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, China*

---

## Abstract

Surface parameterization is of great importance for many applications such as quadrangulation, texture mapping and surface fitting. An important issue for surface parameterization is how to align parametric lines with feature directions. To address this issue, in this paper we first utilize Loop subdivision basis functions and isogeometric analysis (IGA) to calculate eigenfunctions of the secondary Laplace operator (SLO) on triangle meshes. Eigenfunctions are then used for centroidal Voronoi tessellation (CVT) based surface segmentation, and boundaries of the segmented regions are extracted as feature lines which contain concave creases and convex ridges. Along each feature line, adjacent triangles are defined as guidance triangles to parameterize the surface using a constrained cross field method, where feature lines are preserved and aligned to parametric lines. Several examples are presented in the end to verify the robustness of our algorithm.

© 2016 The Authors. Published by Elsevier Ltd.

Peer-review under responsibility of organizing committee of the 25<sup>th</sup> International Meshing Roundtable (IMR25).

**Keywords:** surface parameterization; Loop subdivision; isogeometric analysis; secondary Laplace operator; centroidal Voronoi tessellation; surface segmentation.

---

## 1. Introduction

Surface parameterization computes a one-to-one mapping between a 3D surface mesh and an isomorphic planar patch. It has a variety of applications in different fields, including surface quadrangulation [1,2], texture mapping [3,4] and surface fitting [5]. For parameterization-based quadrilateral (quad) meshing, vector field guided methods such as mixed-integer quadrangulation [6], periodic global parametrization [2] and QuadCover [1] have been developed to generate quality quad meshes. A typical vector field guided method usually consists of three main steps [7]: a cross field is firstly constructed on the input surface which specifies the orientation and size of quad elements; the surface is then partitioned by a set of curves to a topological disk and parametrized into an integer grid map; and finally, a quad mesh can be extracted by tracing the integer parametric lines on the surface. These methods generate curvature-oriented quad meshes by optimizing the cross field which is derived from the principal curvatures. Eigenfunctions of the Laplace-Beltrami operator (LBO) were used in [8] as the guidance to capture major structure features during the

---

\* Corresponding author. Tel.: +1-412-268-5332 ; fax: +1-412-268-3348.

E-mail address: [jessicaz@andrew.cmu.edu](mailto:jessicaz@andrew.cmu.edu)

cross field based global parameterization. Harmonic field based methods [9,10] construct conformal parameterizations with singularities and offer a degree of control over the size and structure of the domain mesh.

Various discretization schemes of the LBO have been proposed on discretized surface meshes [11–13], such as the cotangent scheme [14], Fujiwara's discretization [15], and Mayer's discretization [16]. A discretized LBO with convergent property was recently constructed in [17]. Since eigenfunctions of the LBO are capable of capturing structural features of an object, they have been intensively studied for surface matching and segmentation. The ShapeDNA [18,19] employs level sets of the LBO eigenfunctions for statistical shape analysis. Point clustering [20,21] was used together with the LBO eigenfunctions for surface segmentation. Interactive approaches were presented in [22] to choose eigenfunctions and the Mumford-Shah model was then applied to segment the surface into several components. To detect concavities, the concavity-aware Laplacian method [21] was developed and its eigenfunctions can be used to generate a single segmentation field through the spectral clustering. However, it is hard for the LBO eigenfunctions to detect curvature-related features since they are defined based on the first fundamental form of the surface [23]. Improving upon the LBO, the secondary Laplace operator (SLO) [24] was developed recently based on the second fundamental form of the surface. Curvature related surface features, such as concave creases and convex ridges, can be automatically captured by its eigenfunctions. Computation of the SLO eigenfunctions in [24] utilizes quadrilateral control meshes with Catmull-Clark basis functions [25], which limits its application on triangle meshes.

Given an input triangular mesh, in this paper we first introduce a centroidal Voronoi tessellation (CVT) based surface segmentation method using the SLO eigenfunctions to extract surface features, and then apply the cross field method to generate a feature-aligned surface parameterization. The SLO eigenfunctions are computed using Loop subdivision basis functions and isogeometric analysis. The first several modes are selected for CVT-based surface segmentation, and boundaries of the segmented regions are extracted as feature lines which contain concave creases and convex ridges. Along each feature line, adjacent triangles are defined as the guidance for cross field construction. A constrained surface parameterization is then computed, where feature lines are preserved and aligned to parametric lines. The key contributions of this paper include:

1. Loop subdivision basis functions are utilized together with isogeometric analysis (IGA) to solve the eigenproblem of the SLO over triangular meshes, which is defined based on the second fundamental form of the surface;
2. A CVT-based surface segmentation approach is developed in the eigenfunction space, where the  $L_\infty$  norm is used as the distance measurement. Compared to the  $L_2$  norm distance measurement, the  $L_\infty$  norm distance metric is more robust to identify surface features by taking the dominant feature of the difference vector between a vertex and its associated generator. By considering both the eigenfunction similarity and segmented boundary smoothness, regions surrounded by curvature related features can be segmented while generators are iteratively updated in the eigenfunction space; and
3. Boundaries of the segmented regions are used to define guidance triangles and their guidance directions. The constrained cross field method parameterizes the surface with all feature lines aligned to the parametric lines.

The remainder of this paper is organized as follows. Section 2 talks about the SLO and its eigenfunction computation using Loop subdivision based IGA method. Section 3 discusses CVT-based surface segmentation and cross field-based parameterization. Section 4 shows some results, and Section 5 draws conclusions and points out future work.

## 2. SLO Eigenfunction Computation Using Loop Subdivision Basis Functions

Given an input triangle mesh, we use Loop subdivision basis functions to define a smooth representation of the surface. The SLO eigenfunctions are calculated using the subdivision-based IGA method.

### 2.1. Secondary Laplace Operator

Let  $S = \{\mathbf{x}(u, v), (u, v) \in \mathbb{R}^2\}$  be a smooth and closed parametric surface, where  $(u, v)$  can also be written as  $(u^1, u^2)$  for convenience. The coefficients of the first fundamental form of  $S$  are defined as  $g_{\alpha\beta} = \langle \mathbf{x}_{,u^\alpha}, \mathbf{x}_{,u^\beta} \rangle$  ( $\alpha, \beta = 1, 2$ ), where

$\mathbf{x}_{u^\alpha} = \frac{\partial \mathbf{x}}{\partial u^\alpha}$  and  $\mathbf{x}_{u^\beta} = \frac{\partial \mathbf{x}}{\partial u^\beta}$ . The coefficients of the second fundamental form of  $S$  are defined as  $b_{\alpha\beta} = \langle \mathbf{n}, \mathbf{x}_{u^\alpha u^\beta} \rangle$ , where  $\mathbf{x}_{u^\alpha u^\beta} = \frac{\partial^2 \mathbf{x}}{\partial u^\alpha \partial u^\beta}$  and  $\mathbf{n} = (\mathbf{x}_u \times \mathbf{x}_v) / \|\mathbf{x}_u \times \mathbf{x}_v\|$ . Let  $g = \det[g_{\alpha\beta}]$ ,  $[g^{\alpha\beta}] = [g_{\alpha\beta}]^{-1}$ , and  $[b^{\alpha\beta}] = [b_{\alpha\beta}]^{-1}$ . Given  $f \in C^2(S)$ , the LBO [21,26] acting on  $f$  is defined as

$$\Delta f = \operatorname{div}(\nabla f) = \frac{1}{\sqrt{g}} \left[ \frac{\partial}{\partial u}, \frac{\partial}{\partial v} \right] \left[ \sqrt{g} [g^{\alpha\beta}] [f_u, f_v]^T \right], \quad (1)$$

where  $\nabla$  is the tangential gradient operator given by

$$\nabla f = [\mathbf{x}_u, \mathbf{x}_v] [g^{\alpha\beta}] [f_u, f_v]^T, \quad (2)$$

and  $\operatorname{div}$  is the tangential divergence operator defined by

$$\operatorname{div}(\mathbf{v}) = \frac{1}{\sqrt{g}} \left[ \frac{\partial}{\partial u}, \frac{\partial}{\partial v} \right] \left[ \sqrt{g} [g^{\alpha\beta}] [\mathbf{x}_u, \mathbf{x}_v]^T \mathbf{v} \right]. \quad (3)$$

Improving up the LBO which is defined based on the first fundamental form of the surface [23], the SLO  $\blacktriangle$  is defined based on the second fundamental form of the surface [24]. It is given implicitly as

$$\int_S (h \blacktriangle f + \langle \blacklozenge f, \blacklozenge h \rangle) dA = 0, \quad \forall h \in C^1(S), \quad (4)$$

where the generalized second tangential operator (GSTO)  $\blacklozenge$  can be defined as

$$\blacklozenge f = [\mathbf{x}_u, \mathbf{x}_v] \Phi [b^{\alpha\beta}] [f_u, f_v]^T = g_u^\blacklozenge f_u + g_v^\blacklozenge f_v, \quad (5)$$

with  $g_u^\blacklozenge = \frac{\Phi}{b} (b_{22}\mathbf{x}_u - b_{12}\mathbf{x}_v)$  and  $g_v^\blacklozenge = \frac{\Phi}{b} (b_{11}\mathbf{x}_v - b_{12}\mathbf{x}_u)$ . Different choices of the parameter  $\Phi$  can be used for various applications [24]. Let  $\lambda_L$  and  $\lambda_S$  be the eigenvalues of LBO and SLO respectively, the corresponding eigenfunctions  $f_L$  and  $f_S$  should satisfy

$$\Delta f_L = -\lambda_L f_L \quad \text{and} \quad \blacktriangle f_S = -\lambda_S f_S. \quad (6)$$

Different from the widely employed LBO eigenfunctions, the SLO eigenfunctions can capture the curvature-related surface features [24], which automatically distinguish different components of an object.

## 2.2. Eigenfunction Computation of SLO

Letting  $\{\varphi_i\}_{i=1}^N$  be a set of basis functions defined on the surface, where  $N$  is the vertex number and  $\varphi_i \in C^2(S)$ ,  $f$  can be approximately represented as  $f = \sum_{i=1}^N w_i \varphi_i$ . Plugging  $h = \varphi_j$  ( $j = 1, 2, \dots, N$ ) into Eq. (4), we obtain

$$\sum_{i=1}^N w_i \int_S \langle \nabla \varphi_i, \nabla \varphi_j \rangle dA = \lambda_L \sum_{i=1}^N w_i \int_S \varphi_i \varphi_j dA$$

and

$$\sum_{i=1}^N w_i \int_S \langle \blacklozenge \varphi_i, \blacklozenge \varphi_j \rangle dA = \lambda_S \sum_{i=1}^N w_i \int_S \varphi_i \varphi_j dA$$

for the LBO and SLO, respectively. Letting  $m_{ij}^L = \int_S \langle \nabla \varphi_i, \nabla \varphi_j \rangle dA$ ,  $m_{ij}^S = \int_S \langle \blacklozenge \varphi_i, \blacklozenge \varphi_j \rangle dA$ , and  $c_{ij} = \int_S \varphi_i \varphi_j dA$ , the eigenfunctions of LBO and SLO can be obtained by solving the eigenproblems

$$M^L W = \lambda_L C W \quad \text{and} \quad M^S W = \lambda_S C W, \quad (7)$$

where  $M^L = [m_{ij}^L]$ ,  $M^S = [m_{ij}^S]$ ,  $C = [c_{ij}]$ , and  $W = [w_1, w_2, \dots, w_N]^T$ .

Since SLO is defined based on the second fundamental form of the surface, a high-order representation of the surface is required to compute its eigenfunctions. Catmull-Clark basis functions and quadrilateral control meshes were used in [24] to represent the surface, where surface quadrangulation is required as the preprocessing for input triangle meshes. In this paper, we directly use triangle meshes with Loop subdivision basis functions to represent the surface, and the eigenproblems of LBO and SLO can be solved using the Loop subdivision based IGA method [24,27]. Given a triangle mesh  $K$ , the Loop subdivision scheme subdivides each triangle into four subtriangles in each subdivision step, where vertices of the refined mesh are calculated as the weighted average of vertices of the unrefined mesh. Let us consider the subdivision process from the  $k$ -th level to the  $(k+1)$ -th level, where  $k = 0, 1, \dots$ . The initial mesh  $K$  is taken as the subdivision surface of the 0-th level. Let  $\mathbf{x}_0^k$  be a vertex at level  $k$  with one-ring neighbors  $\mathbf{x}_i^k$  ( $i = 1, \dots, n$ ), where  $n$  is the valence of  $\mathbf{x}_0^k$ . Then the vertex position is updated by

$$\mathbf{x}_0^{k+1} = (1 - n\alpha)\mathbf{x}_0^k + \alpha(\mathbf{x}_1^k + \mathbf{x}_2^k + \dots + \mathbf{x}_n^k), \quad (8)$$

where  $\alpha = \frac{1}{n} \left[ \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{n} \right)^2 \right]$ . Letting  $\mathbf{x}_l^k$  and  $\mathbf{x}_r^k$  be the two wing neighbor vertices of the edge  $[\mathbf{x}_0^k, \mathbf{x}_i^k]$ , the new vertex created on this edge is defined as

$$\mathbf{x}_{0i}^{k+1} = \frac{3}{8}\mathbf{x}_0^k + \frac{3}{8}\mathbf{x}_i^k + \frac{1}{8}\mathbf{x}_l^k + \frac{1}{8}\mathbf{x}_r^k. \quad (9)$$

Note that all the newly generated vertices have a valence of 6, while vertices inherited from the original mesh may have a valence other than 6. The vertex of valence 6 is referred to as a *regular vertex*, and the vertex of valence other than 6 is referred to as an *extraordinary vertex*. The limit surface of the Loop subdivision is  $C^2$ -continuous at regular vertices and  $C^1$ -continuous at extraordinary vertices [28].

To obtain a local parameterization of the limit surface for each triangle in the initial control mesh  $K$ , we choose  $(u, v)$  as two barycentric coordinates in  $(1 - u - v, u, v)$  and define  $\bar{T}$  as

$$\bar{T} = \{(u, v) \in \mathbb{R}^2 : u \geq 0, v \geq 0, u + v \leq 1\}.$$

For a regular surface patch whose corresponding initial triangle has three regular vertices, it can be exactly described by a quartic box-spline via twelve local control vertices and their basis functions. We have

$$\mathbf{x}(u, v) = \sum_{i=1}^{12} N_i(u, v) \mathbf{x}_i, \quad (10)$$

where the basis functions  $N_i$  are given in [28].

For an irregular surface patch whose corresponding initial triangle has one or more extraordinary vertices, the mesh needs to be subdivided recursively until the parameter values of interest are interior to a regular patch according to a fast evaluation strategy [28] under the assumption that any irregular patch has only one extraordinary vertex. Each subdivision of an irregular patch produces three regular subpatches and one irregular subpatch. The piecewise parametric subdomains  $\bar{T}_j^k$  at the subdivision level  $k$  are given as follows:

$$\begin{aligned} \bar{T}_1^k &= \{(u, v) : u \in [2^{-k}, 2^{-k+1}], v \in [0, 2^{-k+1} - u]\}, \\ \bar{T}_2^k &= \{(u, v) : u \in [0, 2^{-k}], v \in [2^{-k} - u, 2^{-k}]\}, \\ \bar{T}_3^k &= \{(u, v) : u \in [0, 2^{-k}], v \in [2^{-k}, 2^{-k+1} - u]\}. \end{aligned}$$

These subdomains can be mapped onto  $\bar{T}$  via the following transformations:

$$\begin{aligned} t_{k,1}(u, v) &= (2^k u - 1, 2^k v), & (u, v) &\in \bar{T}_1^k, \\ t_{k,2}(u, v) &= (1 - 2^k u, 1 - 2^k v), & (u, v) &\in \bar{T}_2^k, \\ t_{k,3}(u, v) &= (2^k u, 2^k v - 1), & (u, v) &\in \bar{T}_3^k. \end{aligned}$$

The entire irregular patch is then defined by its restriction to each regular subpatch

$$\mathbf{x}(u, v) \Big|_{\bar{T}_j^k} = \sum_{i=1}^{12} N_i(t_{k,j}(u, v)) \mathbf{x}_i^{k,j}, \quad j = 1, 2, 3; k = 1, 2, \dots, \quad (11)$$

where  $\mathbf{x}_i^{k,j}$  are properly chosen from the control vertices around the irregular patch at the subdivision level  $k$ , and they define a regular subpatch. Hence, the main task becomes how to compute these control vertices. A Jordan canonical decomposition of the subdivision matrix [28] is used here to speed up the computation in the subdivision process.

For each vertex  $\mathbf{x}_i$  of a control mesh  $K$ , we can associate it with a basis function  $\varphi_i$ , where  $\varphi_i$  is defined by the limit of Loop subdivision for zero values everywhere except at  $\mathbf{x}_i$ , where it is one. Note that the basis  $\varphi_i$  is different from the basis  $N_i$  in Eq. (10).  $\varphi_i$  is a piecewise function whose support covers 2-ring neighboring triangles, whereas  $N_i$  is defined on one triangle only [29]. Using the basis functions  $\{\varphi_i\}_{i=1}^N$ , the limit surface of Loop subdivision can be expressed as  $S = \sum_{i=1}^N \varphi_i(\mathbf{x}_i) \mathbf{x}_i$ . To solve the eigenproblems of Eq. (7), we follow three main steps:

1. Precompute Loop subdivision basis functions and their first derivatives for each patch;
2. Evaluate matrix elements  $m_{ij}$  and  $c_{ij}$  over  $S$  using the 6-point Gauss-Legendre integral formula; and
3. Assemble matrix elements  $m_{ij}$  and  $c_{ij}$  into the eigenproblem system  $MW = \lambda CW$ .

As mentioned above,  $(1 - u - v, u, v)$  are the barycentric coordinates of the generic triangle in the mesh  $K$ . Using this parameterization, our discretized representation of  $K$  is  $K = \bigcup_{\alpha} T_{\alpha}$ ,  $\bar{T}_{\alpha} \cap \bar{T}_{\beta} = \emptyset$  for  $\alpha \neq \beta$ , where  $\bar{T}_{\alpha}$  is the interior of the triangular patch  $T_{\alpha}$ . Each triangular patch is assumed to be parameterized locally as

$$\mathbf{x}_{\alpha} : \bar{T} \rightarrow T_{\alpha}; \quad (u, v) \mapsto \mathbf{x}_{\alpha}(u, v), \quad (12)$$

where  $\mathbf{x}_{\alpha}(u, v)$  is defined by Eqs. (10) and (11). Note that our parameterization has no overlap. Each point  $\mathbf{x} \in K$  has its unique parameter coordinates except at the boundary of each patch. With this parameterization, the matrix element  $c_{ij}$  can be computed as

$$c_{ij} = \int_S \varphi_i \varphi_j dA = \sum_{\alpha} \int_{T_{\alpha}} \varphi_i \varphi_j dA = \sum_{\alpha} \iint_{\bar{T}} \varphi_i(\mathbf{x}_{\alpha}(u, v)) \varphi_j(\mathbf{x}_{\alpha}(u, v)) \sqrt{g} du dv,$$

and the matrix element  $m_{ij}$  is replaced by

$$m_{ij}^L = \int_S \langle \nabla \varphi_i, \nabla \varphi_j \rangle dA = \sum_{\alpha} \int_{T_{\alpha}} \langle \nabla \varphi_i, \nabla \varphi_j \rangle dA = \sum_{\alpha} \iint_{\bar{T}} [\varphi_{iu}, \varphi_{iv}] [g^{\alpha\beta}] [\varphi_{ju}, \varphi_{jv}]^T \sqrt{g} du dv$$

and

$$m_{ij}^S = \int_S \langle \diamond \varphi_i, \diamond \varphi_j \rangle dA = \sum_{\alpha} \int_{T_{\alpha}} \langle \diamond \varphi_i, \diamond \varphi_j \rangle dA = \sum_{\alpha} \iint_{\bar{T}} \Phi^2 [\varphi_{iu}, \varphi_{iv}] [b^{\alpha\beta}] [g_{\alpha\beta}] [b^{\alpha\beta}] [\varphi_{ju}, \varphi_{jv}]^T \sqrt{g} du dv,$$

where  $\varphi_{iu} = \frac{\partial \varphi_i(\mathbf{x}_{\alpha}(u, v))}{\partial u}$  and  $\varphi_{iv} = \frac{\partial \varphi_i(\mathbf{x}_{\alpha}(u, v))}{\partial v}$ . The integration on the triangle  $\bar{T}$  is computed by subdividing the triangle adaptively and then using the Gauss-Legendre quadrature.

We can then assemble  $c_{ij}$  and  $m_{ij}$  into the global matrices  $C$  and  $M$ , and solve the eigenproblems accordingly. Various eigenfunctions of LBO/SLO reflect surface features at different scales. Fig. 1 shows the first four eigenfunctions of the Bunny and  $L$ -shape models, as well as the first five eigenfunctions of the Hook model. To detect both concave creases and convex ridges of an object, we calculate SLO eigenfunctions with  $\Phi = 1$  for all the models in this paper. We can observe that the LBO eigenfunctions follow the main structure of the surface smoothly, but it is insensitive to the variation of surface curvatures. Compared to LBO, the SLO eigenfunctions can detect curvature related features (e.g., the concave creases of the Bunny, and sharp features of the Hook and  $L$ -shape) because it is defined based on the second fundamental form of the surface [24].

### 3. Surface Parameterization with Feature Preservation

After eigenfunction computation, in this section we first apply CVT based surface segmentation to extract feature lines, and then utilize the cross field method to generate a feature-aligned surface parameterization.

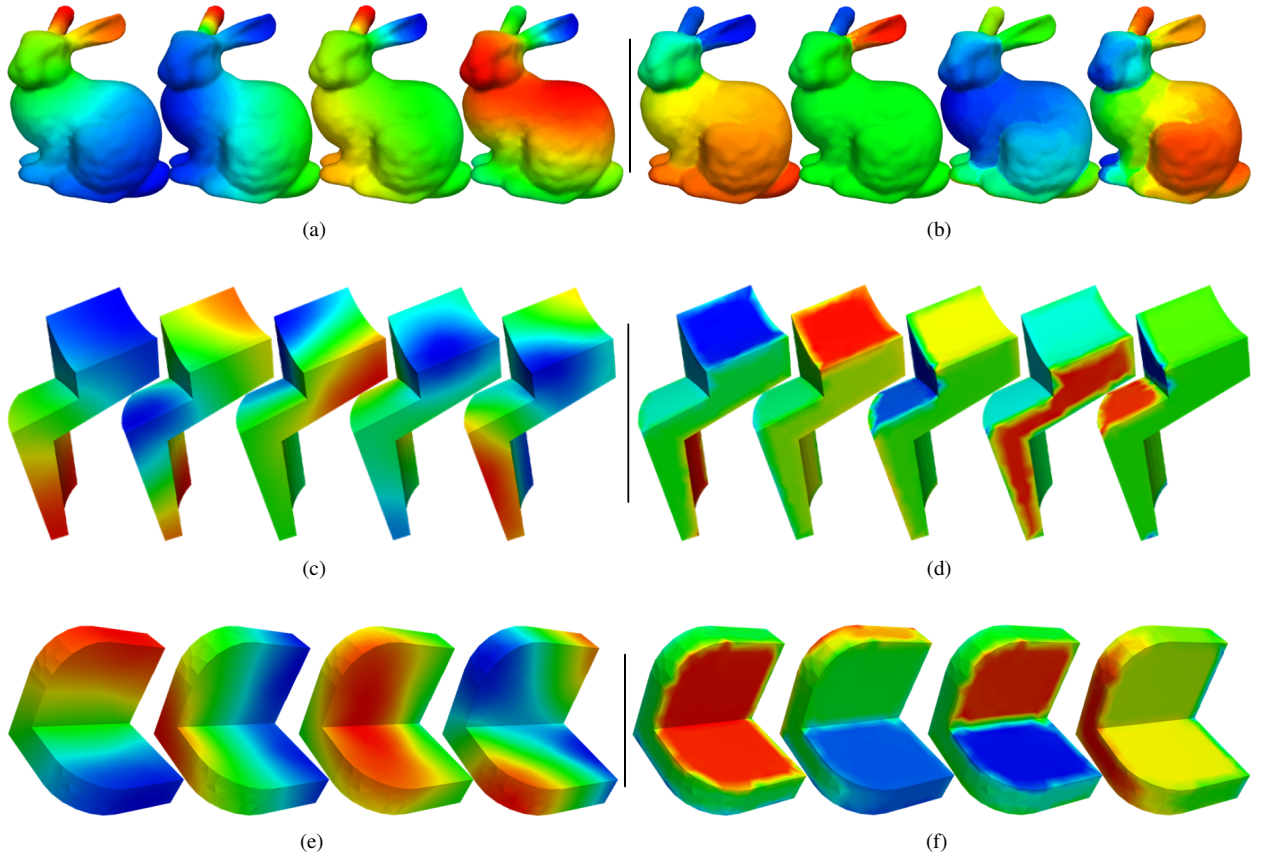


Fig. 1. The first four or five eigenmodes of the LBO (a, c, e) and SLO (b, d, f) for the Bunny (a, b), Hook (c, d) and L-shape (e, f) models.

### 3.1. Feature Extraction via Surface Segmentation

We begin this section by reviewing CVT-based clustering techniques. The CVT scheme has been introduced to various fields and applications, including both image and mesh processing [30–32], which generates an optimal domain partition corresponding to an optimal distribution of generators. Given an input point cloud and  $L$  generators, each point is firstly assigned to its nearest generator with certain distance metric to construct  $L$  non-overlapping Voronoi regions. The centroid of each Voronoi region can be computed by minimizing an energy function  $E$  which measures the clustering similarity. Each generator is iteratively updated to be the centroid of its associated Voronoi region, where a new partition can be computed. Such an algorithm aims at minimizing the energy function  $E$  until each centroid coincides with the corresponding generator. The edge-weighted CVT (EWCVT) model [31,33] can segment images with noise by combining the image intensity information together with the length of cluster boundaries in the edge-weighted clustering energy function. As a follow up, the harmonic EWCVT (HEWCVT) [34] outperforms the classic CVT methods [30,31] by introducing a harmonic form of clustering energy functional to generate stable image segmentation results. Several methods have been proposed to compute the CVT on curved surfaces [35–38]. The CVT energy function from the Euclidean space was further extended to the spherical and hyperbolic spaces in [39], called the universal covering spaces of surfaces. A GPU-based method was proposed in [40] for computing the CVT on the planar domain, leading to a significant speedup over CPU-based methods. CVT-based surface remeshing algorithms [41,42] compute restricted Voronoi diagrams defined as the intersection between the input mesh and a Voronoi diagram. The CVT method [43] can also be used for line segments and graphs. The harmonic boundary-enhanced CVT (HBECVT) [44] extends the HEWCVT-based image segmentation to mesh segmentation in the normal space for automatic polycube construction. In this section, we further extend the HBECVT-based clustering idea to the SLO eigenfunction space for surface segmentation of triangle meshes.

We can map vertices onto a  $p$ -dimensional space by selecting  $p$  eigenmodes of the SLO. Let  $\Psi = \{\psi_i\}_{i=1}^{n_v}$  denote  $p$ -dimensional eigenfunctions for all vertices of the surface mesh with  $\psi_i = (\phi_{i1}, \dots, \phi_{ip})$ , where  $n_v$  is the total number of vertices,  $\psi_i$  represents the assigned  $p$ -dimensional vector of the  $i^{\text{th}}$  vertex  $\mathbf{x}_i$ , and  $\phi_{ij} = f_j(\mathbf{x}_i)$ ,  $j = 1, 2, \dots, p$ ,  $f_j = \sum_k w_k \varphi_k$  is the  $j^{\text{th}}$  eigenfunction of the SLO. Let  $C = \{c_l\}_{l=1}^L$  denote a set of predefined  $p$ -dimensional vectors, which can be initialized by choosing  $L$  random vertices, and we take their  $p$ -dimensional eigenfunction vectors. The Voronoi regions  $V = \{V_l\}_{l=1}^L$  in  $\Psi$  can be obtained by assigning each vertex to the cluster with the closest generator according to the distance metric:

$$V_k = \{\psi_i \in \Psi : \text{dist}(\psi_i, c_k) \leq \text{dist}(\psi_i, c_l), \quad \text{for } l = 1, \dots, L\}, \quad (13)$$

where  $\text{dist}(\psi_i, c_k)$  measures the distance between  $\psi_i$  and  $c_k$ . To measure the distance between vector  $\psi_i$  and generator  $c_k$  in the  $p$ -dimensional eigenspace, we define the  $L_\infty$  norm based distance metric as

$$\text{dist}(\psi_i, c_k) = \sqrt{\|\psi_i - c_k\|_\infty^2 + \lambda \tilde{n}_k(\mathbf{x}_i)}, \quad (14)$$

where the term  $\|\psi_i - c_k\|_\infty = \max\{|\phi_{i1} - c_{k1}|, |\phi_{i2} - c_{k2}|, \dots, |\phi_{ip} - c_{kp}|\}$  measures the distance in the  $p$ -dimensional eigenspace, the boundary-enhanced term  $\tilde{n}_k(\mathbf{x}_i)$  represents the number of vertices that do not belong to the  $k^{\text{th}}$  cluster within the  $\omega$ -ring ( $\omega = 3$  in this paper) neighborhood of  $\mathbf{x}_i$  and  $\lambda$  is a positive weighting factor to balance these two terms. Note that  $\tilde{n}_k(\mathbf{x}_i)$  includes the local neighbouring information of vertex  $\mathbf{x}_i$ . Given any set of generators  $C = \{c_l\}_{l=1}^L$  and any partition  $U = \{U_l\}_{l=1}^L$  of  $\Psi$ , the  $L_\infty$  norm clustering energy function of  $(C; U)$  can be defined as

$$E(C; U) = \sum_{i=1}^n \left( L \left/ \sum_{l=1}^L \text{dist}^{-2}(\psi_i, c_l) \right. \right). \quad (15)$$

The CVT construction can be viewed as an energy minimization process, where the centroid  $c_k^*$  of each cluster  $V_k$  is calculated by minimizing the clustering energy in Eq. (15) with respect to  $c_k$ . Since there is no derivative available by using the  $L_\infty$  norm, we use the Powell method [45] to numerically calculate  $c_k^*$  for each cluster. If the generators of the Voronoi regions  $\{V_l\}_{l=1}^L$  coincide to their corresponding centroids, i.e.,  $c_l = c_l^*$  for  $l = 1, \dots, L$ , then we call such Voronoi tessellation  $\{V_l\}_{l=1}^L$  a CVT. Otherwise, we set  $c_l = c_l^*$  for  $l = 1, \dots, L$  to start a new iteration by Eq. (13). In our implementation,  $\{c_l^*\}_{l=1}^L$  and  $\{c_l\}_{l=1}^L$  are updated iteratively until the energy variation is less than a predefined threshold. Based on the  $L$  non-overlapping clusters of surface vertices, we then construct the boundaries of each cluster to obtain the final surface segmentation. Fig. 2(b) shows the CVT-based surface segmentation result of the Hollow-cylinder model with four clusters by using the first three SLO eigenmodes (shown in Fig. 2(a)), where neighbouring clusters are rendered with different colors. From the segmentation result, we can observe that regions surrounded by concave creases and convex ridges are well segmented since SLO eigenfunctions are sensitive to the variation of surface curvatures. Based on the segmentation results, boundaries of each cluster are extracted as the feature lines, which will be preserved during the following surface parameterization process.

### 3.2. Surface Parameterization

Using feature lines extracted from surface segmentation, we now build a constrained surface parameterization via the cross field method, where we aim to align feature lines to parametric directions. Along each feature line, we find its adjacent triangles and define them as the guidance triangles. For each guidance triangle, we can calculate its four perpendicular guidance directions by using the edge direction and normal vector. Note that the guidance triangles are used as the constraints to provide guidance for the cross field construction. For the Hollow-cylinder model in Fig. 2, the feature lines are extracted via boundary extraction and marked as green lines in Fig. 2(c). For each blue guidance triangle, we calculate its guidance directions through the direction of the feature edge and the normal vector, see the result in the red window of Fig. 2(c).

For the given triangle mesh  $K$ , the cross field of each triangle  $i$  is represented by an angle  $\theta_i$ . Using the guidance triangles with guidance directions defined by feature lines, a cross field can be obtained by minimizing a smoothness energy function [6]



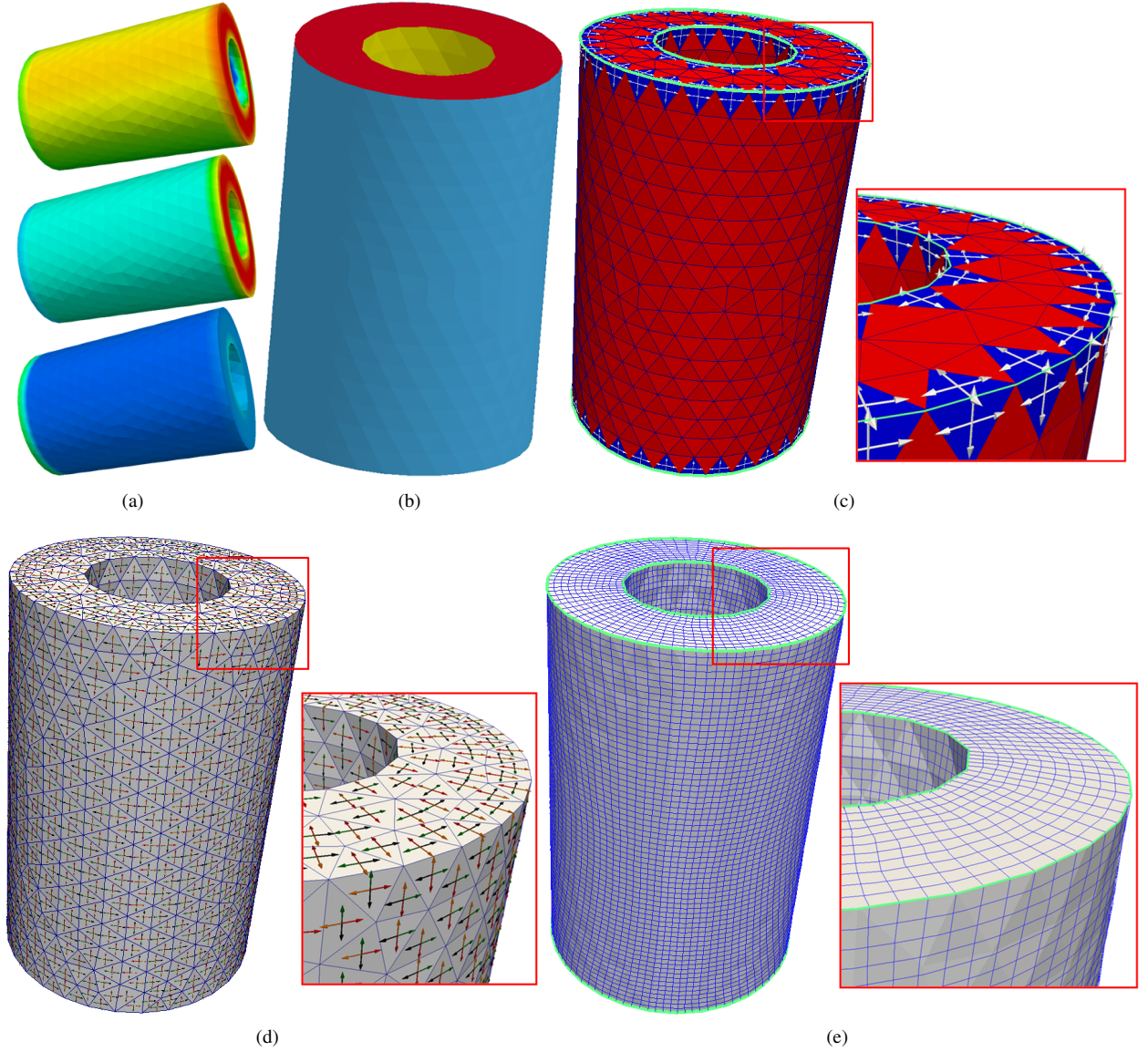


Fig. 2. Hollow-cylinder model. (a) Modes 1-3 of the SLO; (b) CVT-based surface segmentation; (c) feature lines (green), guidance triangles (blue) and their guidance directions; (d) the built smooth cross field (four arrows in each triangle); and (e) surface parameterization with parametric lines.

$$\Gamma_0 = \sum_{e_{ij} \in \mathcal{E}} (\theta_i + \kappa_{ij} + \frac{\pi}{2} p_{ij} - \theta_j)^2, \quad (16)$$

where  $\mathcal{E}$  contains all edges,  $e_{ij}$  is the edge shared by triangle  $i$  and  $j$ ,  $\kappa_{ij}$  is the angle between reference edges  $e_i$  and  $e_j$ , and  $p_{ij}$  represents the integer period jump cross the edge  $e_{ij}$ . The mixed-integer solver [6] is used to solve the minimization problem. Fig. 2(d) shows the calculated cross field of the Hollow-cylinder. Four arrows for each triangle represent the four directions of the cross field that are perpendicular or parallel with each other. The surface is then partitioned into a disk-like planar region, where all the singularities are positioned on the boundary of the planar region. The surface parameterization is computed as a solution to the constrained minimization problem [6]



$$\sum_{T_i \in M} A_i (\|\nabla u - \beta \mathbf{u}_i\|^2 + \|\nabla v - \beta \mathbf{v}_i\|^2) \rightarrow \min, \quad (17)$$

where  $\beta$  is used to control the spacing of the parametric lines (in this paper we choose  $\beta = 0.2$ ),  $A_i$  is the area of triangle  $i$ ,  $\mathbf{u}_i$  and  $\mathbf{v}_i$  are two directions chosen from the four directions of the cross field in triangle  $i$ , and  $(u, v)$  are the parametric coordinates. The constraints imposed on  $(u, v)$  values correspond to transitions across seams: we want the match across seams to be the same as for the guiding cross field. Apart from transition constraints, constraints are imposed on vertices of each feature line. For two vertices of each sharp edge, we compare the edge direction with  $\mathbf{u}_i$  and  $\mathbf{v}_i$  and set constraints to ensure that these two vertices have the same  $u$  or  $v$  coordinates. For vertices along the feature lines, we also set integer constraints according to the  $(u, v)$  coordinates of the corresponding feature edges so that the generated parametric lines will align with the feature edges exactly. In addition, all singularities are constrained to be at integer locations in the parametric domain, which ensures all singularities are at quad corners and all quad edges are matched across disk-like planar regions. The quadrangulation can be generated by tracing the integer parametric lines on the surface. Fig. 2(e) shows the surface parameterization result of the Hollow-cylinder with parametric lines (marked as blue lines), which align to the feature lines detected by surface segmentation.

**Remark 3.1.** Various eigenfunctions of the SLO can capture different curvature related surface features. When we use multiple eigenmodes of the SLO for the surface segmentation, the  $L_\infty$  norm distance metric computes the distance from a vertex to a generator by taking the dominant feature of the difference vector in the  $p$ -dimensional eigenfunction space. By considering both the eigenfunction similarity and segmentation boundary smoothness in the boundary-enhanced distance metric, our CVT-based surface segmentation can extract curvature related surface features with smooth boundaries. Our surface parameterization generates feature-aligned quadrangulation results by applying two constraints. Firstly, the pre-defined directions in guidance triangles follow feature lines and they are used as constraints for the cross field construction in Eq. (16). Secondly, integer constraints are applied to vertices along the feature lines during the minimization process in Eq. (17), yielding feature-aligned surface parameterization.

#### 4. Results and Discussion

We have applied the presented algorithms to several datasets, including the Bunny (Fig. 3), Teddy (Fig. 4), Hook (Fig. 5) and  $L$ -shape (Fig. 6). For each model, we calculated SLO eigenfunctions, CVT-based surface segmentation and surface parameterization with feature preservation. All results were computed on a PC equipped with a 2.93 GHz Intel X3470 CPU and 8GB of Memory. Statistics of all tested models are given in Table 1, where we show a summary of the number of singularities and the computational time for each model. For CVT-based surface segmentation, we need to select SLO eigenfunctions and define two parameters:  $L$ , the number of clusters; and  $\lambda$ , the weighting parameter that balances the clustering energy and the boundary-enhanced energy.

Fig. 3 shows the surface segmentation and parameterization results of the Bunny model. We segment the surface into 5 clusters by using the first eigenfunction of the SLO. As shown in Fig. 3(a), it is obvious that our CVT-based method segments the concave and convex regions well since the SLO eigenfunctions are sensitive to curvature-related surface features. Based on the segmentation result, we extract boundaries of each cluster as the feature lines and define the corresponding guidance triangles with their guidance directions for the cross field construction. Compared to the unconstrained surface parameterization, our constrained parameterization aligns the resulting parametric lines to the feature lines, see the comparison in the red windows of Fig. 3(b, c). Fig. 4 shows the results of the Teddy model, where Modes 1-4 of the SLO are used to segment it into 6 clusters. Since multiple eigenmodes are used, here we generate CVT-based segmentation results using the  $L_2$  and  $L_\infty$  norm distance metrics respectively under the same initialization; see Fig. 4(b, c). We can observe that the segmentation using the  $L_\infty$  norm distance measurement segments surface features better, where the head and legs are well separated from the body. Fig. 4(d) shows the final feature-aligned surface parameterization result.

We also applied our algorithms to several CAD models with sharp features. Figs. 5 and 6 show results of the Hook and  $L$ -shape models. For the Hook model, we segment the surface into 6 clusters by using Modes 1-5 of the SLO. Fig. 5(a) and (b) show the CVT-based segmentation results using the  $L_2$  and  $L_\infty$  norm distance metrics respectively in the 5-dimensional eigenspace under the same initialization. For the  $L$ -shape model, we segment the surface into 6 clusters

Table 1. Statistics of all the tested models.

Model (vertices, elements)	Modes	$\lambda$	Number of clusters	Number of singularities	$T_E$ (s)	$T_S$ (s)	$T_P$ (s)	$T_T$ (s)
Hollow-cylinder (927, 1,854)	1-3	0.02	4	0	6.7	1.4	5.6	13.7
Bunny (14,076, 28,148)	1	0.01	5	42	93.6	18.8	76.7	189.1
Teddy (6,104, 12,204)	1-4	0.01	6	34	41.1	8.3	33.6	83.0
Hook (2,256, 5,128)	1-5	0.02	6	16	15.5	3.2	12.9	31.6
L-shape (538, 1,072)	1-4	0.02	6	12	4.5	0.9	3.7	9.1

Note: (vertices, elements) - the number of vertices and triangles in the input mesh;  $T_E$  - time for eigenfunction computation;  $T_S$  - computational time for CVT-based surface segmentation;  $T_P$  - computational time for surface parameterization; and  $T_T$  - the total computational time. (Time unit: second).

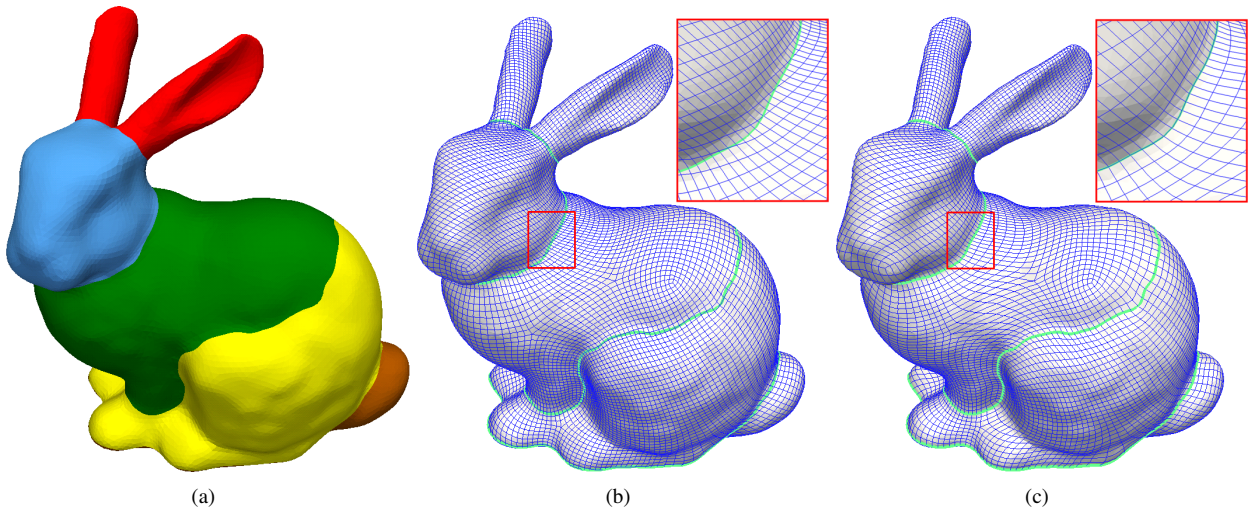


Fig. 3. Bunny model. (a) Surface segmentation result from the first eigenfunction of the SLO; (b) unconstrained surface parameterization; and (c) constrained surface parameterization.

by using Modes 1-4 of the SLO. We also generate CVT-based segmentation results using the  $L_2$  and  $L_\infty$  norm distance metrics respectively; see Fig. 6(a, b). We can observe that regions with sharp features are not well segmented when the  $L_2$  norm is used although each SLO eigenfunction can detect the curvature related features at various scales, while the  $L_\infty$  norm distance metric performs better to detect regions surrounded by curvature-related feature lines. Fig. 5(c) shows the surface parameterization result of the Hook with all sharp features aligned to parametric lines. As shown in Fig. 6(c), feature lines of the L-shape are extracted as the cluster boundaries and blue triangles along the extracted feature lines (green lines) are selected as the guidance triangles to build the cross field. The resulting parametric lines align well to the feature lines after the constrained surface parameterization process, see the result in Fig. 6(d).

**Limitations.** Although eigenfunctions of the SLO can well detect curvature related surface features, selecting proper eigenmodes for our CVT-based surface segmentation is heuristic and it needs some user interactions. Since our constrained surface parameterization algorithm requires aligning parametric lines with integer constraints along

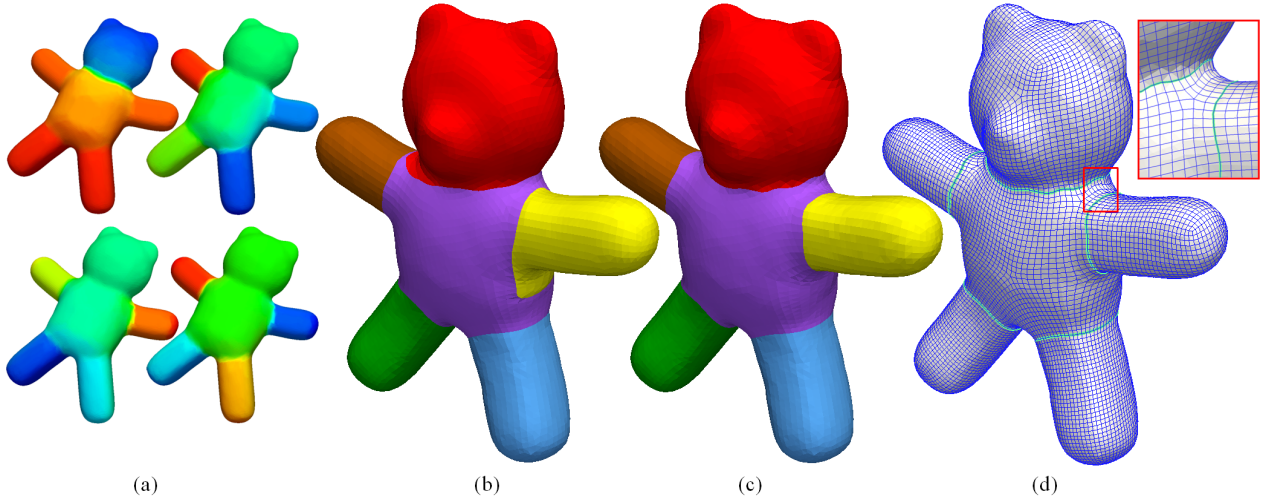


Fig. 4. Teddy model. (a) Modes 1-4 of the SLO; (b) CVT-based surface segmentation using the  $L_2$  norm distance metric; (c) CVT-based surface segmentation using the  $L_\infty$  norm distance metric; and (d) surface parameterization.

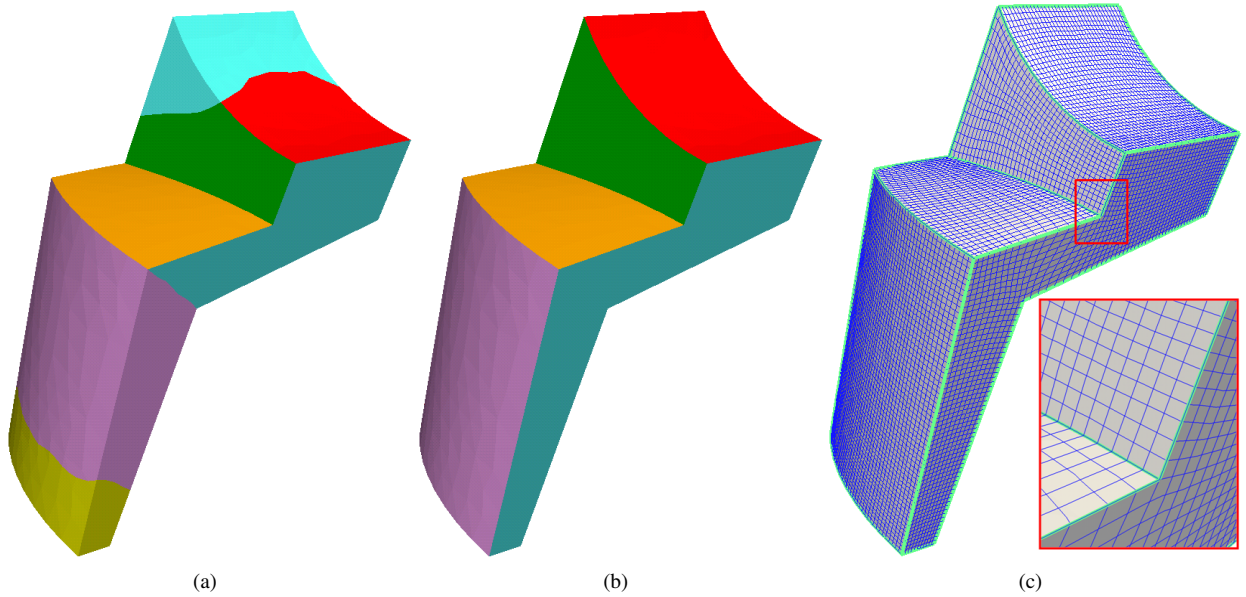


Fig. 5. Hook model. (a) CVT-based surface segmentation using the  $L_2$  norm distance metric; (b) CVT-based surface segmentation using the  $L_\infty$  norm distance metric; and (c) surface parameterization.

the feature lines, large distortion may be introduced sometimes. For this situation, we need to improve the quality of the resulting parameterization using smoothing and/or optimization methods.

## 5. Conclusions and Future Work

In this paper, we have developed a feature-aligned surface parameterization algorithm with the help of the SLO eigenfunctions and Loop subdivision basis functions. To capture curvature related surface features, we first utilize the Loop subdivision based IGA method to calculate eigenfunctions of the SLO over triangle surfaces. We then employ the CVT-based surface segmentation in the eigenfunction space to extract feature lines which contain concave creases and convex ridges. The  $L_\infty$  norm is adopted here as the distance measurement. A constrained cross field method

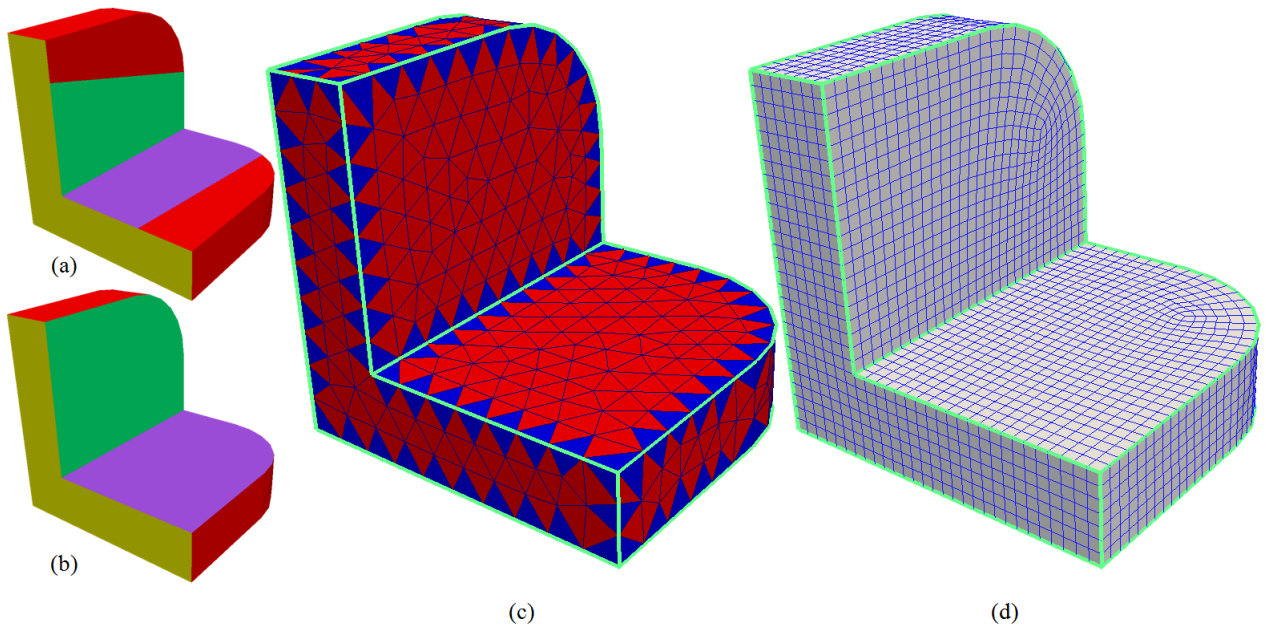


Fig. 6. L-shape model. (a) CVT-based surface segmentation using the  $L_2$  norm distance metric; (b) CVT-based surface segmentation using the  $L_\infty$  norm distance metric; (c) feature lines and the corresponding guidance triangles; and (d) surface parameterization.

is developed for surface parameterization, where feature lines are preserved and aligned to parametric lines. In the future, we plan to study the discretization scheme of the SLO over triangle meshes, and extend the presented method to construct quad meshes, T-meshes and analysis-suitable T-splines for IGA applications.

## Acknowledgment

The work of K. Hu and Y. Zhang was supported in part by PECASE Award N00014-16-1-2254 and NSF CAREER Award OCI-1149591. X. Li and G. Xu were supported in part by NSFC Fund for Creative Research Groups of China under the grant 11321061.

## References

- [1] F. Kälberer, M. Nieser, K. Polthier, QuadCover-surface parameterization using branched coverings, *Computer Graphics Forum* 26 (2007) 375–384.
- [2] N. Ray, W. C. Li, B. Lévy, A. Sheffer, P. Alliez, Periodic global parameterization, *ACM Transactions on Graphics* 25 (2006) 1460–1485.
- [3] L. Wang, X. Gu, K. Mueller, S. Yau, Uniform texture synthesis and texture mapping using global parameterization, *The Visual Computer* 21 (2005) 801–810.
- [4] B. Lévy, Constrained texture mapping for polygonal meshes, *ACM SIGGRAPH* (2001) 417–424.
- [5] B. Jüttler, A. Felis, Least-squares fitting of algebraic spline surfaces, *Advances in Computational Mathematics* 17 (2002) 135–152.
- [6] D. Bommes, H. Zimmer, L. Kobbelt, Mixed-integer quadrangulation, *ACM Transactions On Graphics* 28 (2009) 1–10.
- [7] D. Bommes, B. Lévy, N. Pietroni, E. Puppo, C. Silva, M. Tarini, D. Zorin, Quad-mesh generation and processing: A survey, *Computer Graphics Forum* 32 (2013) 51–76.
- [8] T. Liao, G. Xu, Y. Zhang, Structure-aligned guidance estimation in surface parameterization using eigenfunction-based cross field, *Graphical Models* 76 (2014) 691–705.
- [9] S. Dong, P. Bremer, M. Garland, V. Pascucci, J. C. Hart, Spectral surface quadrangulation, *ACM Transactions on Graphics* 25 (2006) 1057–1066.
- [10] Y. Tong, P. Alliez, D. Cohen-Steiner, M. Desbrun, Designing quadrangulations with discrete harmonic forms, *Symposium on Geometry Processing* (2006) 201–210.
- [11] G. Xu, Q. Pan, C. Bajaj, Discrete surface modelling using partial differential equations, *Computer Aided Geometric Design* 23 (2006) 125–145.
- [12] G. Xu, Q. Zhang, A general framework for surface modeling using geometric partial differential equations, *Computer Aided Geometric Design* 25 (2008) 181–202.

- [13] G. Xu, Consistent approximations of several geometric differential operators and their convergence, *Applied Numerical Mathematics* 69 (2013) 1–12.
- [14] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, *Visualization and mathematics III* (2003) 35–57.
- [15] K. Fujiwara, Eigenvalues of Laplacians on a closed Riemannian manifold and its nets, *Proceedings of the American Mathematical Society* 123 (1995) 2585–2594.
- [16] U. F. Mayer, Numerical solutions for the surface diffusion flow in three space dimensions, *Computational and Applied Mathematics* 20 (2001) 361–379.
- [17] X. Li, G. Xu, Y. Zhang, Localized discrete Laplace-Beltrami operator over triangular mesh, *Computer Aided Geometric Design* 39 (2015) 67–82.
- [18] M. Reuter, S. Biasotti, D. Giorgi, G. Patané, M. Spagnuolo, Discrete Laplace-Beltrami operators for shape analysis and segmentation, *Computers & Graphics* 33 (2009) 381–390.
- [19] M. Reuter, Hierarchical shape segmentation and registration via topological features of Laplace-Beltrami eigenfunctions, *International Journal of Computer Vision* 89 (2010) 287–308.
- [20] R. Liu, H. Zhang, Segmentation of 3D meshes through spectral clustering, in: *Pacific Conference on Computer Graphics and Applications*, 2004, pp. 298–305.
- [21] H. Wang, T. Lu, O. Au, C. Tai, Spectral 3D mesh segmentation with a novel single segmentation field, *Graphical Models* 76 (2014) 440–456.
- [22] J. Zhang, J. Zheng, C. Wu, J. Cai, Variational mesh decomposition, *ACM Transactions on Graphics* 31 (2012) 21:1–21:14.
- [23] G. Xu, Discrete Laplace-Beltrami operators and their convergence, *Computer Aided Geometric Design* 21 (2004) 767–784.
- [24] T. Liao, X. Li, G. Xu, Y. Zhang, Secondary Laplace operator and generalized Giaquinta-Hildebrandt operator with applications on surface segmentation and smoothing, *A Special Issue of SIAM Conference on Geometric & Physical Modeling 2015 in Computer Aided Design* 70 (2016) 56–66.
- [25] X. Wei, Y. Zhang, T. Hughes, M. Scott, Truncated hierarchical Catmull-Clark subdivision with local refinement, *Computer Methods in Applied Mechanics and Engineering* 291 (2015) 1–20.
- [26] B. Lévy, Laplace-Beltrami eigenfunctions towards an algorithm that understands geometry, *IEEE International Conference on Shape Modeling and Applications* (2006) 13–21.
- [27] Q. Pan, G. Xu, G. Xu, Y. Zhang, Isogeometric analysis based on extended Loop's subdivision, *Journal of Computational Physics* 299 (2015) 731–746.
- [28] J. Stam, Fast evaluation of Loop triangular subdivision surfaces at arbitrary parameter values, *SIGGRAPH 98 Proceedings*, CD-ROM supplement, Orlando (1998).
- [29] C. L. Bajaj, G. Xu, Anisotropic diffusion of surfaces and functions on surfaces, *ACM Transactions on Graphics* 22 (2003) 4–32.
- [30] Q. Du, M. Gunzburger, L. Ju, X. Wang, Centroidal Voronoi tessellation algorithms for image compression, segmentation, and multichannel restoration, *Journal of Mathematical Imaging and Vision* 24 (2006) 177–194.
- [31] J. Wang, L. Ju, X. Wang, An edge-weighted centroidal Voronoi tessellation model for image segmentation, *IEEE Transactions on Image Processing* 18 (2009) 1844–1858.
- [32] Q. Du, M. Gunzburger, L. Ju, Advances in studies and applications of centroidal Voronoi tessellations, *Numerical Mathematics: Theory, Methods and Applications* 3 (2010) 119–142.
- [33] Y. Cao, L. Ju, Q. Zou, C. Qu, S. Wang, A multichannel edge-weighted centroidal Voronoi tessellation algorithm for 3D super-alloy image segmentation, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 17–24.
- [34] K. Hu, Y. Zhang, Image segmentation and adaptive superpixel generation based on harmonic edge-weighted centroidal Voronoi tessellation, *The Special Issue of CompIMAGE'14 in Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 4 (2016) 46–60.
- [35] D. Cohen-Steiner, P. Alliez, M. Desbrun, Variational shape approximation, *ACM Transactions on Graphics* 23 (2004) 905–914.
- [36] P. Alliez, É. C. de Verdière, O. Devillers, M. Isenburg, Centroidal Voronoi diagrams for isotropic surface remeshing, *Graphical Models* 67 (2005) 204–231.
- [37] S. Valette, J. M. Chassery, R. Prost, Generic remeshing of 3D triangular meshes with metric-dependent discrete Voronoi diagrams, *IEEE Transactions on Visualization and Computer Graphics* 14 (2008) 369–381.
- [38] X. Wang, X. Ying, Y. Liu, S. Xin, W. Wang, X. Gu, W. Mueller-Wittig, Y. He, Intrinsic computation of centroidal Voronoi tessellation (CVT) on meshes, *Computer-Aided Design* 58 (2015) 51–61.
- [39] G. Rong, M. Jin, L. Shuai, X. Guo, Centroidal Voronoi tessellation in universal covering space of manifold surfaces, *Computer Aided Geometric Design* 28 (2011) 475–496.
- [40] G. Rong, Y. Liu, W. Wang, X. Yin, X. Gu, X. Guo, GPU-assisted computation of centroidal Voronoi tessellation, *IEEE Transactions on Visualization and Computer Graphics* 17 (2011) 345–356.
- [41] D. Yan, B. Lévy, Y. Liu, F. Sun, W. Wang, Isotropic remeshing with fast and exact computation of restricted Voronoi diagram, *Computer graphics forum* 28 (2009) 1445–1454.
- [42] D. Yan, W. Wang, B. Lévy, Y. Liu, Efficient computation of clipped Voronoi diagram for mesh generation, *Computer-Aided Design* 45 (2013) 843–852.
- [43] L. Lu, B. Levy, W. Wang, Centroidal Voronoi tessellations for line segments and graphs, *Technical Report*, INRIA-ALICE, 2009.
- [44] K. Hu, Y. Zhang, Centroidal Voronoi tessellation based polycube construction for adaptive all-hexahedral mesh generation, *Computer Methods in Applied Mechanics and Engineering* 305 (2016) 405–421.
- [45] M. J. Powell, An efficient method for finding the minimum of a function of several variables without calculating derivatives, *The Computer Journal* 7 (1964) 155–162.